Connecting the Risk Decision Authority Criteria (RDAC) to its Environment

Ben Calloni, PhD, CISSP, P.E., Lockheed Martin Fellow, Software Security
Jess Irwin, Senior Manager System Engineering, Raytheon Company
Gordon Uchenick, Coverity
W. Mark Vanfleet, Global Network Vulnerability Analyst, DOD

April 19, 2011

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **19 APR 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**
4. TITLE AND SUBTITLE **Connecting the Risk Decision Authority Criteria (RDAC) to its Environment**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Lockheed Martin,Cherry Hill,NJ,08002**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES **Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License**		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**29**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Objective

- This presentation extends RDAC **architectural analysis** down to embedded computing platforms and up to major programs of record. This presentation ties together:
  - Hardware/Firmware Risk
  - Software Risk
  - Obsolescence Risk
  - Management of these risks via RDAC
  - Risk Management at the Program of Record Level

RDAC: Risk Management Framework

- RDAC: **Risk Decision Authority Criteria**
- Originally created to guide accreditation of CDS solutions in a Secret and Below Initiative (SABI) environment
  - Management of Residual Risk
- Usage has expanded over time to include accreditation of System of Systems solutions
  - **Common Vulnerabilities** found in prior system accreditations
  - **Common Threats** that challenge mission capability
  - **Analysis** that system **architecture** addresses the vulnerabilities and adequately resists the threats

Risk Management

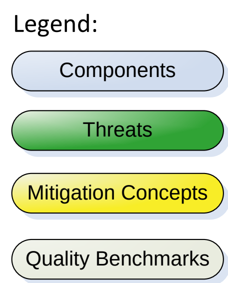
- Impossible technically and financially to counter all known risks
- Mitigate the known risks we can’t counter
- Understand the residual risk after known risks are countered or mitigated
  - Determine if system architecture is sufficiently robust with respect to the residual risk
- What about the risks we don’t know or can’t anticipate?
  - Determine if system architecture is sufficiently robust to withstand additional risks
- *Obsolescence is a significant technical and financial threat not often addressed during original system risk analysis*
  - Obsolescence events typically cost more than the original implementation, certification, and accreditation combined

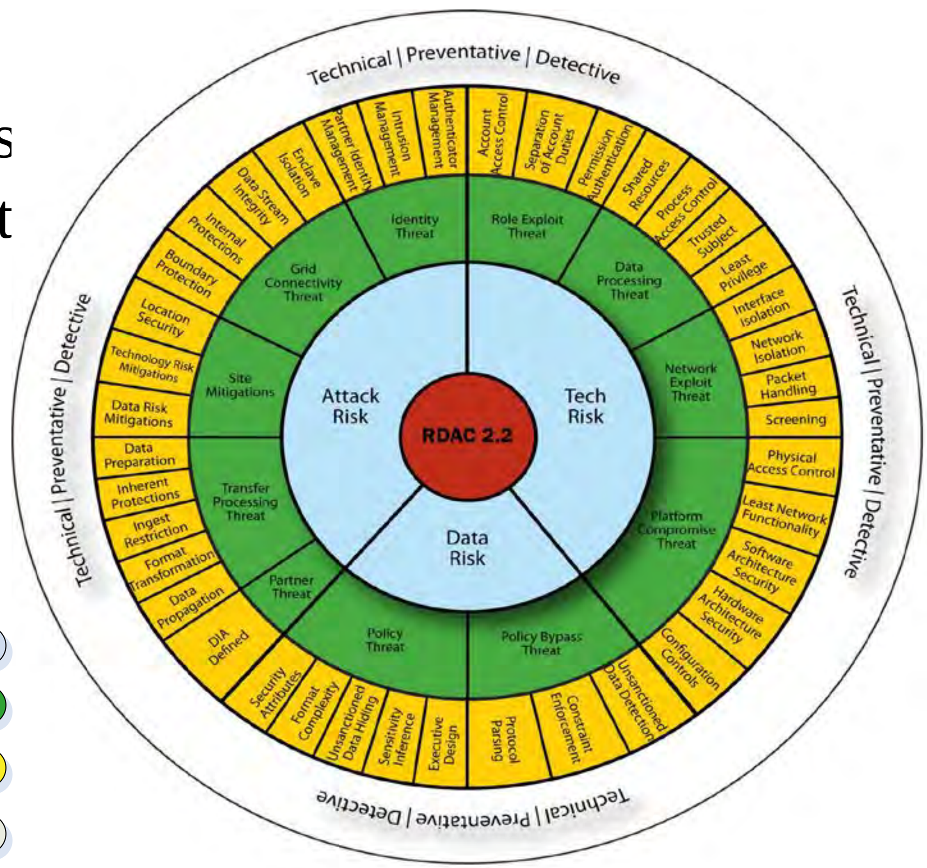
System Life Total Cost of Ownership

- Implementation
- Certification / Accreditation
- Deployment
- **Operations & Maintenance**
- **Technology Refresh**
- **Growing Attack Surface over time**
- **Obsolescence Events**

RDAC Manages Three Kinds of Risks

- Technical Risk
  - Data Processing and Platform Compromise
  - Role and Network Exploit
- Data Risk
  - Security Policy Completenes
  - Security Policy Enforcement
- Attack Risk
  - Identity Threat
  - Connectivity Threats
  - Physical Threats
  - Transfer Threats





Additional Risks to be Managed

- Stronger Adversaries
  - Increased capability of nation-state adversaries
  - Increased capability of sub-national and transnational adversaries
- Increased Attack Surface
  - Larger and more complex systems
  - Everything is connected
    - *Enables world wide remote adversaries*
    - *Mobile personal devices based on commodity software becoming the warfighter's interface of choice*
- Extended System Lifetimes
  - Feature creep complication or invalidating original system security architecture
  - Reuse in unanticipated environments
- ***Yesterday's code was not intended to counter today's threats***

Obsolescence Risk

- COTS hardware platforms become unavailable because of market influences
- COTS software platforms become unavailable for the same reasons
- “Murphy” Obsolescence:
  - Hardware and software do not become obsolete at the same time
  - COTS hardware and software each become obsolete at the worst possible time
  - The version of COTS software that was accredited will not run on any COTS hardware you can still buy
- Murphy was an optimist:
  - Ashton Carter, USD AT&L: “We’re asking you to do more without more.”
    - “And make it network-ready”
    - “And make it Multi-level Secure”
    - “And implement it as a field upgrade”
- ***AND MAKE IT HAPPEN NOW!***

Using RDAC to Manage the Other Risks

- And also mitigate Obsolescence Risk!
- Two pronged approach:
  - Decompose large and complex security policies into small and simple independent components
    - Sufficiently large and/or complex policies may require formal verification (more about this later)
  - Architect systems such that association of data and metadata is incorruptible and verifiable

Managing Development Risk

- Can’t manage development risk without understanding
- Data -> Information -> Knowledge -> Understanding
- Data -> Information (Analysis)
  - Static Analysis of code, data structures, information flow, and requirements
    - Coverity, Code Sonar, Code Surfer, Understand of C++, …
    - University of Kentucky Requirements Analysis Tools
- Information -> Knowledge (Reduction)
  - Consistent interpretation of information and requirements across platforms, analysts, organizations, and decision makers
    - Metadata Tagging (GPS, TOD, Mission Phase, Classification, Release-ability) of all data streams
    - Architectural Analysis Tools and consistent use of Data Dictionary
    - RDAC, Architecture and Requirements Analysis
- Knowledge -> Understanding (Visualization)
  - Mitigation of Risk within Architecture
  - Minimization of total cost of ownership
    - Affordable recovery from zero-day defects and obsolescence events
- **Risk mitigation during development requires continuous review with updated understanding of evolving risks and technology**

Managing Operational Risk

- Can't manage operational risk without understanding
- Data -> Information -> Knowledge -> Understanding
- Data -> Information (Analysis)
  - Consistent interpretation of information across platforms, analysis, organizations, and decision makers
  - Metadata associated with information to give context, classification, and release-ability
- Information -> Knowledge (Reduction)
  - Data and metadata organized so the right questions can be asked
  - Architectural analysis to discover accidental data and control coupling
- Knowledge -> Understanding (Visualization)
  - Actionable assessments
- **Risk mitigation during operation requires continuous monitoring with updated understanding of evolving risks and technology**

Tools Enable Understanding

- Can’t understand risk without tools
  - Commonly exploited vulnerabilities are caused by tangled architectures, lack of standards, and common programming flaws
- Architecture Analysis Tools
  - Visualize and verify compliance with architectural design goals
- Static Analysis Tools
  - Verify correctness and standards compliance of code
- Dynamic Analysis Tools
  - Verify correctness of concurrency models

Decomposition of Large Complex Policies

- Implement the desired policy while simultaneously eliminating possibility of internal inconsistencies
- Provide an infrastructure with three characteristics
  - Integrity of components can’t be compromised
  - Information flow among components is controlled
  - Data is securely associated with its metadata at its source
- Create a flexible policy enforcement framework
  - React quickly to changes in operational requirements
  - Adapt to obsolescence events at reasonable cost

Association of Data and Metadata

- Cross domain guards no longer change the sensitivity of data
  - Redaction rule authors can’t anticipate all use contexts
  - Even perfectly designed and implemented redaction is vulnerable to aggregation
- Cross domain guards become simple information flow reference monitors
  - More resistant to operational risks and obsolescence risks

Planning for Assurance

- Plan for assurance of reference monitors during the entire system life
- Systems evolve with each technical refresh
- Risk management strategy and assessment of that strategy must also evolve in lock-step with the system
- Perform RDAC analysis at each stage of system life to prevent costly security shortcomings later

Example Reference Monitors

1. CDS: movement of data and metadata between domains based on releaseability policy for data and metadata
    a. Data and Metadata Fusion
    b. Data and Metadata Extraction
2. Encryption: data at rest (AT), over the air (IA)
3. Metadata Tagging, binding data to its properties
    a. Mission Phase, GPS, Track, Time/Date
    b. Sensor characteristics
    c. National and classification markings of subject and object
4. Mapping of classification and releaseability markings between nations

Reference Monitor Characteristics: ***NEAT***

- **N**on-bypass-able
  - Information flows only along the paths intended by the system designer and there are no unintended paths
- **E**valuateable
  - Observes Principle of Least Privilege in all aspects
- **A**lways-invoked
  - Policy of Reference Monitor type is correctly enforced each and every time the reference monitor is invoked
- **T**amper-proof
  - Cyber Hard infrastructure and resource management

Infrastructure Characteristics: ***TIME***

- **T**ype safety
  - Sustain service in the face of faults, errors, failures, hostile, and unforeseen use cases
- **I**nfiltration
  - External subjects should NOT have influence over local resources/objects
- **M**ediation
  - Trusted subjects that have access to multiple information flows and/or multiple critical functions should only allow use of the information flow and critical functions by authorized subjects
- **E**xfiltration
  - Internal subjects should NOT have influence over external resources/objects

Non-Iterative Process

- When we try to bolt on security at the end
- **Measures quality of the system only after everyone thinks the job is done**
  - *All the time is gone*
    - (and the system was probably late)
  - *All the money has been spent*
    - (and the system was probably over budget)
- **Obsolescence events may force C&A to start over from the very beginning**

Non-Iterative Process

- Performing RDAC only during original system C&A is reactive
  - Even with requirement analysis traceability tools
    - Requirement satisfaction analysis becomes obsolete over time
  - Even with static analysis tools
    - Fixing *this* often breaks *that*
    - Same code, different platform forced by hardware obsolescence can make original risk analysis irrelevant

Iterative Process

- ***Iteratively* using requirements traceability tools and static analysis tools is proactive**
  - Over time, even innocuous defects become vulnerabilities
  - Ensures continual completeness of the system security solution
  - Ensure continual effectiveness of the solution
  - Discovers voids and defects at the most cost effective time
    - *Prevents "Enhancing this broke that"*
  - Positive effect on total system assurance over the entire system life cycle

Extend RDAC Downward to Embedded System Components

- Embedded component specific system risks
  - Increased hardware complexity (i.e., multi-core)
  - Provenance of COTS firmware
  - OS evolution from bare metal to kernels to RTOS to Linux
  - Increased embedded system scope and complexity
- Applying RDAC to embedded system risks
  - Provenance of active elements
    - Can we trust that integrated memory controller/PCIE Bridge/Network Interface chip?
    - What do we really know about that TCP offload engine?
  - Global hardware and software supply chains
    - Is the original architecture still valid?
    - Have new vulnerabilities been introduced?

Extend RDAC Upward to Programs of Record

- Must consider both operational risk and obsolescence risk
- *Iteratively* performing risk analysis manages obsolescence
    - Reimplementation is implementation at the most expensive stage possible
    - Flexible embedded system frameworks and robust binding of metadata enable adaptation, extension, and interconnection instead of reimplementation

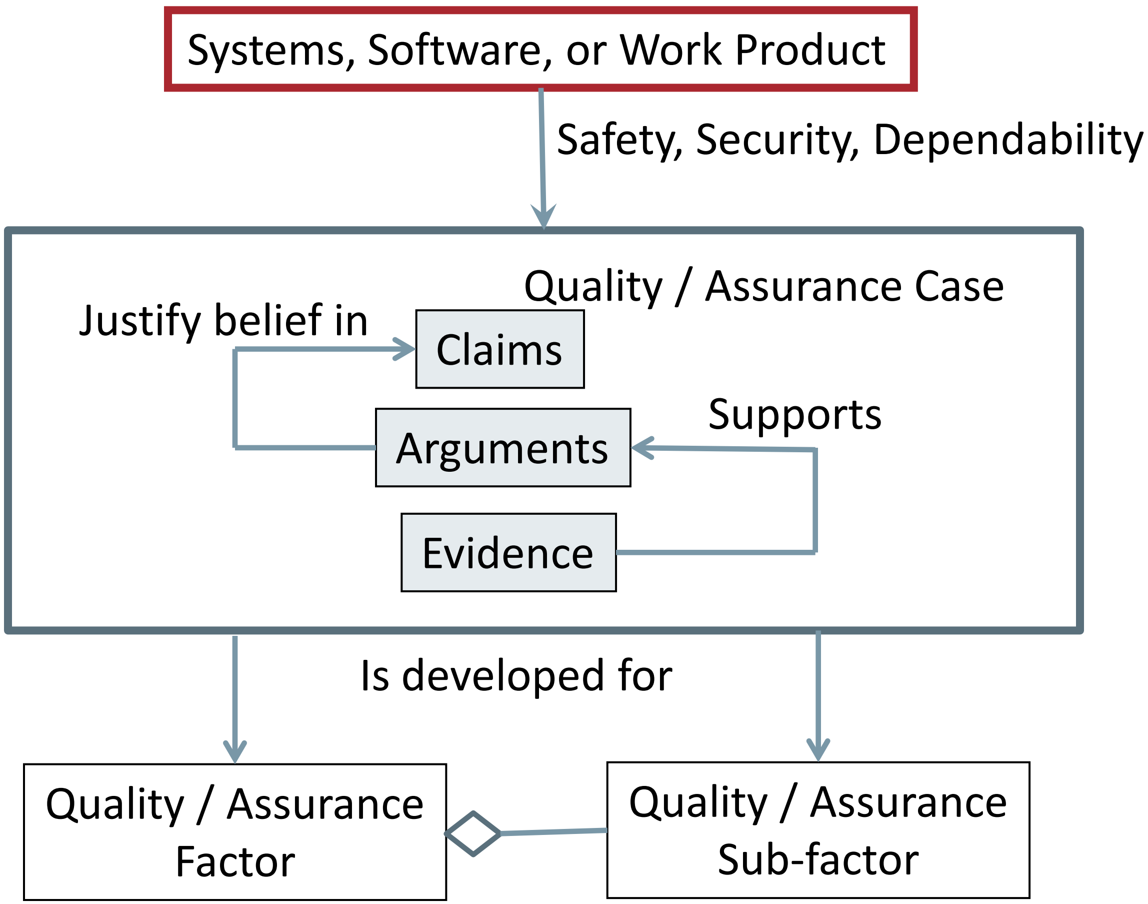
Applying RDAC to Program of Record Risks

- Iterative RDAC process during operation is applicable to all platforms and systems
- RDAC should be adopted as applicable to each platform or system
  - Within capabilities of current management personnel
  - Adoption manages cost and schedule impact of obsolescence events
    - Example: Upgrade of a single line of code can require six aircraft for six months of retesting and can cost up to $6 million

Applying MILS / OSA to RDAC

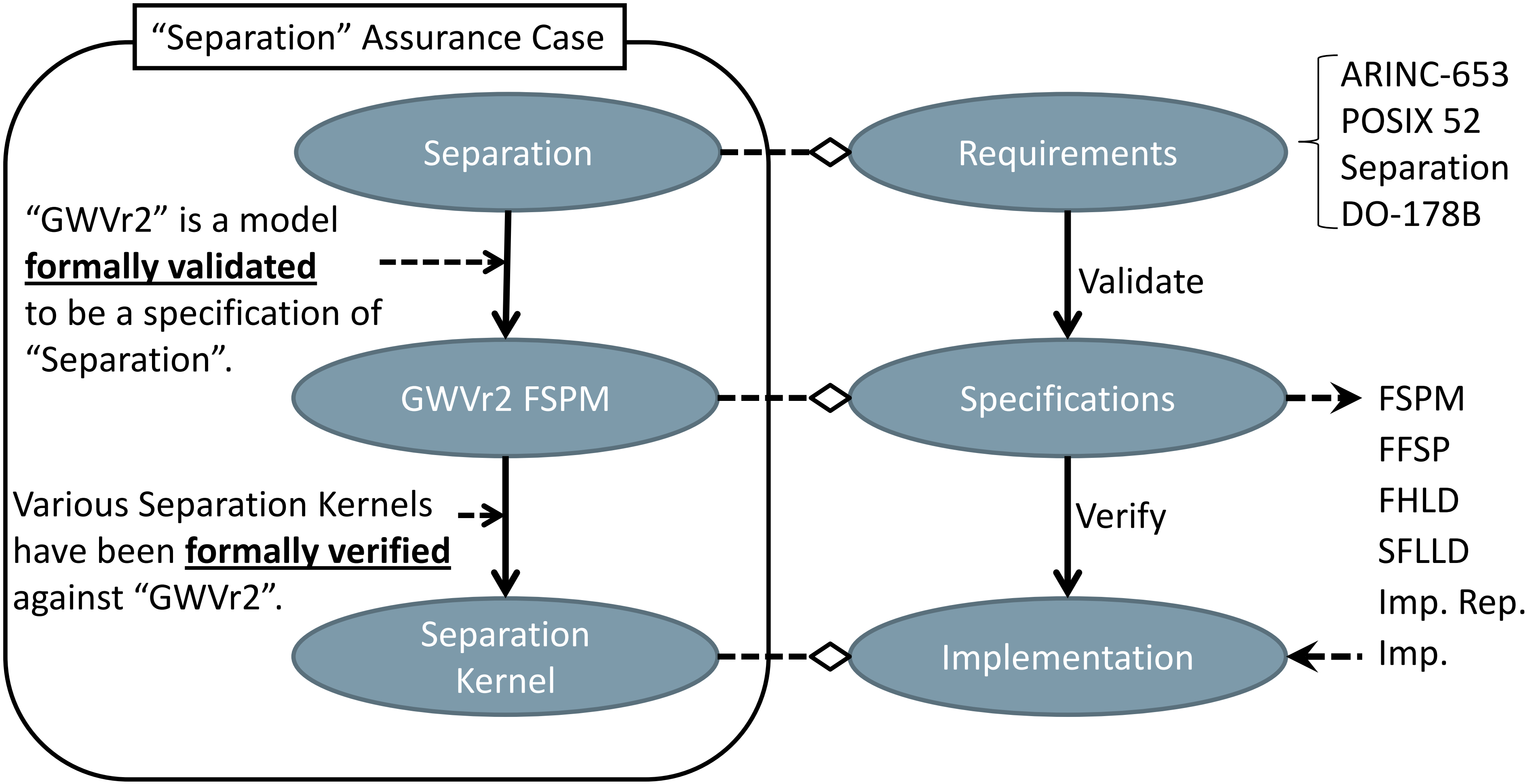
- MILS: Multiple Independent Levels of Security
- OSA: Open System Architecture
- MILS / OSA
  - Leverages separation, damage limitation, periods processing, and controlled information flow
  - Isolates applications from technology and implementations
  - Converts Security Information / Architecture into Security Awareness and Understanding

ISO/IEC/IEEE 15026 Assurance Case



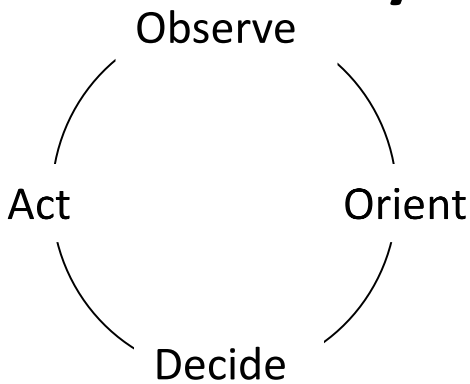
- Attributes
  - Clear, Simple
  - Consistent
  - Complete
  - Comprehensible
  - Defensible
  - Bounded
  - Life Cycle
  - Testable
  - Open Interface
- Dependability
  - Sustain service in the face of faults, errors, failures, hostile, and unforeseen use cases

Formal Validation and Verification of Avionics Systems



"GWVr2" was formally **validated** to provide "Separation" by formally demonstrating that an application level Reference Monitor that is non-bypass-able and tamper-proof can be constructed on an ARINC 653 Separation Kernel. - Proof presented in 2004 ACL2 Workshop.

Survivability



✓**CONFIDENTIALITY**
- Critical Data **PROTECTED**

✓**INTEGRITY**
- Free of Unauthorized Manipulation

✓**AUTHENTICATION**
- Identity Confirmed

✓**AUTHORIZATION**
- Privilege Confirmed
- Mutual Suspicion

(Reduced access based on authentication uncertainty)

✓**NON-REPUDIATION**
- Proof of Data Origin & Delivery

✓**AVAILABILITY**
- Critical functions **READY**

✓**SAFETY**
- Determinism
- Reliability
- Independence

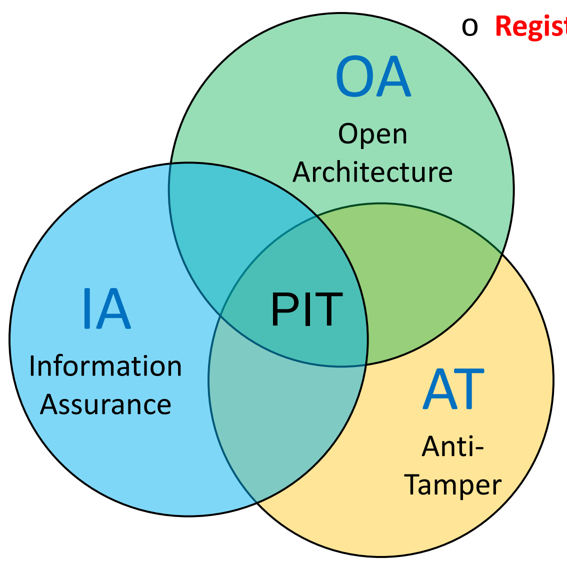
✓**DESIGNATE KEY INFORMATION EXCHANGES**
- **Standardize** similar areas at Enterprise level across Primes
- Blue force tracking, strike, mission planning , weather, …

✓**MODULARITY & VISIABILITY**
- **Design** for change and affordable technology refreshes
- **Minimize** attack surface
- **Design** for recovery and adaptation against Zero-day Defects

✓**RE-USEABLE COMPONENTS**
- Commercial based standards (POSIX, Open GL) - **unmodified**
- Published standards (IEEE 1394, 802.11) - **unmodified**
- Established proprietary standards (USB, Blue Ray) - **unmodified**

✓**INTEROPERABILITY & SECURITY (CJCSI 6212.01E)**
- Global Network Information Enterprise Architecture
- Support for Distributed degree of trust systems

✓**ENABLING ENVIRONMENTS**
- Infrastructure and Enterprise API's **Separable**
- **Decouple** data producers and consumers (cloud computing)
- **Register** data grams and data streams within metadata registry



✓**DETERRENCE**
- Undesirable Consequences
- Strength of Mechanism

✓**PREVENTION**
- Defense in Depth
- Obfuscation

✓**DETECTION**
- Visual, Alarm, Loss of Function, Attestation
- Monitoring

✓**RESPONSE**
- Destruction, Disabling, Zeroization
- Adaption

Web Resources

- Coding Standards and Practices
  - http://www.cert.org/secure-coding/scstandards.html
  - http://cwe.mitre.org/
- National Vulnerability Databases
  - http://web.nvd.nist.gov
  - http://cve.mitre.org/
- DHS Pocket Guides for Security
  - https://buildsecurityin.us-cert.gov/swa/pocket_guide_series.html
- SEI Software Assurance Curriculum
  - http://www.sei.cmu.edu/library/abstracts/reports/10tr005.cfm
- Risk Management Framework
  - http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf